REMARKS

Claims 3-9, 11, 13-23, 26, 30-46, 49, 53, 54, 56-66, 70-79, 81, and 84-101 are pending in this application. Applicant has amended claims 3, 32, 54, 84, and 93 to more particularly point out and distinctly claim attributes of the software redirection driver. Support for these amendments can be found in the claims (*e.g.*, original claims 5 and 66) and the text (*e.g.*, Applicant's Specification, page 4, lines 3-9; Figures 4 and 9, page 15, lines 11-23; and page 8, line 13 – page, line3.) No new matter has been introduced by way of these amendments.

Acknowledgement of Rule 132 Declarations

Applicant thanks the Examiner for his explicit acknowledgement (in Office Communication, Paper No. 20071101) of Applicant's Rule 132 Declaration, filed on June 21, 2007, and Applicant's Supplemental Rule 132 Declaration, filed on September 28, 2007.

35 U.S.C. § 103 Rejections

The Examiner has rejected claims 3-9, 11, 13-15, 30-40, 53-54, 56-66, 71-79, 81, and 84-101 under 35 U.S.C § 103(a) as obvious over Harish et al., U.S. Patent No. 5,940,850 (hereinafter "Harish"). The Examiner appears to have also rejected claims 16-23, 31, 41-46, 49, 65, and 70 under 35 U.S.C § 103(a) as being obvious over Harish in view of Kobayashi et al., U.S. Patent No. 5,437,018 (hereinafter "Kobayashi").

Applicant respectfully traverses all these rejections for the reasons discussed in detail below with respect to both the original and amended claims as indicated.

Rejections over Harish

The Examiner appears to be asserting that Harish's described techniques, which include virtual memory management techniques for loading "dynamic data" stored in a read-only memory into a random access memory when the dynamic data is being modified, somehow teach, suggest, or motivate Applicant's claimed techniques for "automatically preserving an original state of a computer system upon rebooting" by "read[ing] data from and writ[ing] data to a redirected data area (a redirected space) when a storage access request is received *that would*

otherwise alter the state of an area of the storage device that has been designated as protected." (Applicant's Specification, page 2, lines 19-21 (emphasis added).) More specifically, the Examiner appears to equate Harish's virtual memory manager with the "software redirection driver" recited by each of Applicant's claims. (Office Action dated February 7, 2008, page 4.)

Applicant respectfully disagrees that Harish's virtual memory manager can be equated with Applicant's "software redirection driver," as recited by each of Applicant's independent claims 3, 32, 54, 72, 79, 84, and 93, prior to amendment. For a number of reasons, a virtual memory manager is simply not the same as, and cannot be equated to, a "software redirection driver" as recited by Applicant's claims. First, virtual memory managers and device drivers, such as Applicant's recited "software redirection driver" are dedicated to performing different functions in a typical computing system. Virtual memory managers process *memory accesses*, by translating logical memory addresses into physical memory addresses. (Harish, column 4, lines 6-10 (hereinafter in column:line format).) They also are responsible for allowing code to refer to more memory than is physically present in the computing system – hence the use of the term "virtual" in the name virtual memory manager. (Harish, 3:57-61.) Drivers, in contrast, process *input/output device accesses*, such as the read/write operations that occur with respect to disk drives or other storage devices. (See, *e.g.*, Applicant's Specification, page 8, lines 13-22.) Drivers are often configurable in a computing system, and different ones can be loaded for use with an operating system, even after the computing system has been deployed. (See also, *e.g.*, Specification, page 4, lines 3-9.)

The distinction between memory access and input/output device access is reinforced by the fact that such accesses typically relate to, and interact with, different components in a computer system, as shown in Figure 1 of Harish. Specifically, Figure 1 of Harish shows a RAM 106, a ROM 104, and an I/O Controller 108, all as separate components. Consequently, virtual memory managers and drivers are typically implemented as distinct and different portions of a computing operating system. (Compare Silberschatz and Galvin, "Operating System Concepts," 4th Ed., 1994, p. 59 ("main memory management") with p. 60 ("I/O system management").)

Second, virtual memory managers are typically implemented at least in part with specialized hardware that implements memory address translation efficiently. (Silberschatz and Galvin, p. 247 ("Selection of a memory-management scheme ... depends on many factors, especially on the *hardware* design of the system. Each algorithm requires its own hardware support." (Emphasis original.)) For example, simple virtual to physical address translation may be implemented by way of a hardware memory management unit comprising one or more relocation registers. (Silberschatz and Galvin, p. 255.) Alternatively, page-based address translation, such as that described by Harish, is typically implemented by way of a hardware memory management unit comprising a set of associative registers (also called translation lookaside buffers) that contain page table entries. (Silberschatz and Galvin, p. 273.) In any case, hardware support for virtual memory management is critical in order to provide for the efficient operation of a computer system. Without such hardware support, any virtual memory scheme would incur tremendous overhead, as every memory access (*e.g.*, to fetch an instruction, to store a data value, etc.) would need to be translated in software. Accordingly, the virtual memory manager described by Harish is implemented at least in part in hardware and cannot be fairly called a *software* redirection driver as recited by Applicant's claims.

Third, there is nothing in Harish that describes "loading" of the memory manager. The memory manager is already present.

Accordingly, Applicant's recited "loading a software redirection driver into a volatile memory" (independent claims 3, 32, 84, 72, 93) "a software redirection driver loaded into a volatile memory" (independent claim 54), or "a software redirection driver, installed into a volatile memory" (independent claim 79) are not taught, motivated or suggested by the memory manager described by Harish.

Nonetheless, in interests of furthering prosecution, and not to be interpreted as agreement with the Examiner's position, Applicant has amended some of the independent claims. In particular, claim 3, as amended, recites, "loading a software redirection driver into an *input/output driver hierarchy* loaded in a volatile memory of the computer system, wherein *the software redirection driver is an input/output driver*; under control of code of the software redirection driver, *redirecting input/output requests* by ..." Claim 32, as amended, recites,

"loading a software redirection driver into an *input/output driver hierarchy* loaded in a volatile memory of the computer system, wherein *the software redirection driver is an input/output driver*; under control of code of the software redirection driver, *redirecting input/output requests* by ..." Claim 54, as amended, recites, "a software redirection driver that *redirects input/output requests*, loaded into an *input/output driver hierarchy* loaded in a volatile memory of the computer system when the system is booted from a powered-down state, wherein *the software redirection driver is an input/output driver*, including code that, when executed, ..." Claim 84, as amended, recites, "loading a software redirection driver into an *input/output driver hierarchy* loaded in a volatile memory of the computer system, wherein *the software redirection driver is an input/output driver*; under control of code of the software redirection driver, *redirecting input/output requests* by ..." Claim 93, as amended, recites, "loading a software redirection driver into an *input/output driver hierarchy* loaded in a volatile memory of the computer system, wherein *the software redirection driver is an input/output driver*; under control of code of the software redirection driver, *redirecting input/output requests* by ..." (Emphasis added throughout.)

These amendments emphasize 1) that the software redirection driver is an I/O driver – not a memory manager, 2) that it is loaded into the hierarchy of drivers that handle the I/O of the system, and 3) that it redirects input/output requests – not memory accesses. In addition, the "intercepting," and other acts/functions recited in claims thereafter, <u>are performed by the driver</u> – not as a result of a page fault or "write-access exception" – which is an error condition raised by the operating system itself. (See, Harish, 4: 15-27.)

As discussed above, Harish's virtual memory manager differs from Applicant's software redirection driver in various ways. First, Harish's virtual memory manager is not a "software redirection *driver*" by virtue of the fact that it does not function by "redirecting input/output requests" as recited in Applicant's claims. Instead, Harish's virtual memory manager processes memory accesses, which are not the same as input/output requests. In particular, input/output requests are related to the operation of input/output devices, including disks, keyboards, display screens, etc. Accordingly, Harish does not teach, suggest, or motivate

"redirecting input/output requests" performed by a "software redirection driver," as recited by Applicant's claims.

Second, Harish's virtual memory manager is not a "*software* redirection driver" as recited by Applicant's claims. More specifically, Harish nowhere describes whether the virtual memory manager is implemented in software, hardware, or both. To the extent that the Examiner is asserting that Harish describes a software implementation, it is respectfully suggested that the Examiner is relying on facts not of record. If anything, based on the known virtual memory manager design principles discussed above and described in a known operating system textbook, it is only reasonable to assume that for efficiency reasons, Harish's virtual memory manager relies at least in part on specialized hardware to perform address translation and other functions related to its operation. Thus, Harish does not teach, suggest, or motivate a "*software* redirection driver."

Third, Harish's virtual memory manager is not loaded "into a driver hierarchy" of the computer system as recited by Applicant's claims. Harish nowhere describes a "driver hierarchy." The Examiner appears to address the concept of a "driver hierarchy" in his analysis of claim 5. (Office Action, p. 8, citing Harish 4:57-5:18.) However, the cited passage merely describes the initialization of the virtual memory manager, including the initialization of page table entries that refer to physical addresses in ROM. (*Id.*) Initialization of a page table is not the same as a driver hierarchy. (Compare to Applicant's Specification, page 4, lines 3-9 and page 15, lines 11-23, describing a driver hierarchy or driver chain in a typical operating system.) Thus, Harish does not teach, suggest, or motivate a software redirection driver loaded "into a driver hierarchy" as recited by Applicant's claims.

Applicant has not amended independent claims 72 and 79. These claims recite additional aspects nowhere taught, suggested, or motivated by Harish. In particular, claim 72, without amendment, recites, "installing the software redirection driver before the device driver in a calling sequence of the operating system, so that the operating system invokes the redirection driver in response to receiving a request to access the storage device." The Examiner appears to apply his analysis of claim 3 to claim 72, and does not address this aspect of claim 72 with any specificity. The Examiner is not free to simply ignore the language recited in the claim.

Applicant has reviewed Harish and can find no teaching, suggestion, or motivation of a "software redirection driver" installed "before the device driver in a calling sequence of the operating system," as recited by claim 72. Therefore, Applicant respectfully submits that the Examiner has failed to put forth a *prima facie* case and respectfully requests the Examiner to provide support for his rejection of this claim language.

In addition, claim 79, without amendment, recites, "an available space table; a protected space redirection table that is used to designate protected locations on the storage device that are to be protected from modification; an unprotected space table that is used to designate unprotected locations on the storage device that can be altered." Again, the Examiner appears to apply his analysis of claim 3 to claim 79, and does not address this aspect of claim 79 with any specificity. With reference to other claims, the Examiner has equated Harish's ROM with the protected space table and Harish's RAM with the unprotected space table and with the redirection table. Arguably ROM cannot act as the *protected* table as that term is used in Applicant's specification – to protect data that would otherwise be altered. (See, Applicant's Specification, page 2, lines 15-21) Rather, Harish temporarily stores data in ROM that is intended to be modified ("dynamic data," such as variable data that changes for an application or for a user) – it is moved to RAM when needed, because RAM is the scare resource. (Harish, 2:10-35, Abstract.) Even assuming, *arguendo*, that ROM could equate to Applicant's recited protected table, nowhere does the Examiner address the presence of all three tables. Further, Harish does not appear to describe the use of all three tables, as recited by claim 79. Applicant respectfully submits that the Examiner has failed to put forth a *prima facie* case and respectfully requests the Examiner to provide support for his rejection of this claim language.

Accordingly, Harish does not teach, suggest, or motivate one or more aspects of each of independent claims 3, 32, 54, 72, 79, 84, and 93, and thus all of the pending dependent claims, at least by virtue of their dependencies.

Rejections over Harish in view of Kobayashi

Applicant finds the discussion and use of Kobayashi unclear. In particular, the heading on page 13 of the Office Action indicates that Examiner is using Kobayashi in

combination with Harish to reject various of Applicant's dependent claims. (Office Action, p. 13.) However, on pages 14 and 15 of the Office Action, the Examiner mentions "the scheme disclosed by Hansen." In interest of furthering prosecution, and because Hansen is nowhere else mentioned in the Office Action, Applicant assumes that the Examiner is referring to Harish whenever he mentions "Hansen."

In addition, page 14 of the Office Action includes a lengthy discussion of Kobayashi, referring apparently to claim language that does not match the language of any of the claims as currently presented for examination. (See, *e.g.*, Office Action, p. 14, lines 17-19 ("such that the request transparently accesses the current redirected location instead of the original location").) It appears that much of this text may have been inserted from a prior Office Action, and thus the "motivation" for combining the references is incomprehensible in view of the subject matter of Harish (memory management), which is unrelated to storage systems, let alone storage systems with sectors, clusters, etc. Therefore, Applicant respectfully submits that the Examiner has failed to put forth a *prima facie* case with regard to claims 16-23, 41-46, 49, and 70 requests the Examiner to clarify the rejection of the claims over Harish in view of Kobayashi.

Dependent Claims

Each of the dependent claims depends on one of independent claims 3, 32, 54, 72, 79, 84, and 93 addressed above. Therefore, each dependent claim incorporates one or more aspects not taught, suggested, or motivated by the corresponding cited references. As such, each of the dependent claims is not anticipated or obvious for at least the reasons discussed above, with respect to its corresponding ancestor independent claim.

In addition, many of the dependent claims recite additional aspects that are not taught, suggested, or motivated by the cited references. For example, with respect specifically to claims 57 and 71, Applicant notes that the Examiner has rejected these claims based upon **previously deleted claim 27**. Accordingly, the Examiner has not presented a *prima facie* case for rejecting these claims.

Also, with respect specifically to claims 7, 34, and 59, the Examiner indicates that the RAM in Harish provides that "the determined location in the redirected space resides in the storage device." Since the "storage device" contains the protected space (which the Examiner equates to the Harish ROM), it cannot also contain the "determined location in the redirected space," because a ROM cannot also contain a RAM. Thus, the Examiner's interpretation appears to be nonsensical, and the Examiner has not presented a *prima facie* case for rejecting claims 7, 34, and 59.

Other dependent claim rejections present additional issues. Applicant notes for the record that all such assertions are traversed and reserves the right to further present arguments regarding the Examiner's statements about what is known in the art or taught by the cited references at a later time, should such become necessary. Specifically, no waiver (legal, factual, or otherwise), implicit or explicit, is hereby intended.

Conclusion

In the event the Examiner disagrees with Applicant or finds minor informalities, Applicant respectfully requests a telephone interview to discuss the Examiner's issues and to expeditiously resolve prosecution of this application. Accompanying this Amendment is an Applicant Initiated Interview Request Form in the event the Examiner does not agree that the claims are allowable over the cited references. Applicant's representative can be contacted at (206) 622-4900.

In closing, applicant respectfully submits that all of the pending claims are allowable and respectfully requests the Examiner to enter these amendments and to reconsider this application and its timely allowance. The Director is authorized to charge any additional fees due by way of this Amendment, or credit any overpayment, to our Deposit Account No. 19-1090. Again, applicant's representative thanks the Examiner for his prompt and courteous attention.

Respectfully submitted,

SEED Intellectual Property Law Group PLLC

Ellen M. Bierman

Ellen M. Bierman
Registration No. 38,079

EMB:asl

Enclosures:

Applicant Initiated Interview Request Form
Information Disclosure Statement
Cited Reference

701 Fifth Avenue, Suite 5400
Seattle, Washington 98104
Phone: (206) 622-4900
Fax: (206) 682-6031

1197277_1.DOC